現在、私たちは最近多くのコードを書いています。

実際に構築しようとしているものを簡単に確認しましょう。構築しようとしている最終的なアプリは次のようになります。

注目したいのは、場所と天気データを取得するロード画面と、天気情報をユーザーに表示する場所画面があるという事実です。このレッスンでは、小さなスピナーを追加して、ロード画面とロケーション画面の間の遷移を改善します。コードに関しては、これまでに実装したものです。前回のレッスンでは、ここでnetworking.dartヘルパークラスを作成しました。このクラスでは、このクラスを初期化するときにURLを渡し、そのURLを使用してhttp get要求を作成します。

次に、応答が良好かどうか、つまり200で要求が正常に受信されたかどうかを確認します。そして、応答の本文を取得し、jsonDecodeを使用してそのデータをデコードし、必要な場所に送り返します。

そのため、このネットワークヘルパーの柔軟性が非常に高いことに注目してください。

気象データを処理するだけでなく、ネットワーク関連のあらゆる処理を実行できます。さらに、ネットワーク機能がさらに必要な場合に、将来的に他の目的に使用できるように、その機能を独自のカスタムロケーションに分割しました。

getLocationDataメソッドでは、ユーザーの現在の位置を取得し、次にnetworkHelperを使用してその位置の天気データを取得しています。そして、別の場所で使用できる気象データオブジェクトが作成されます。

それで、私たちがそれを必要とする他のどこかが私たちの位置スクリーンです。

ユーザーに画面上に表示できるように、これらのテキストウィジェットにそのデータを配置する必要があります。

そのデータをロケーション画面に渡すために、以前に学んだことを使用する必要があります。これは、ナビゲーターを使用して画面にルートをプッシュすることです。

したがって、コンテキストはコンテキストとして保持されます。

ルートはMaterialPageRouteを使用して作成され、ビルダーはコールバックを予期します。

コンテキストの入力があり、ロケーション画面であるかを表示したい画面を返します。

そしてもちろん、location\_screen.dartファイルをここにインポートしない限り、ロケーション画面については知りません。これで、ロケーション画面のルート設定ができました。そのため、アプリを再度実行してホットリスタートすると、この黒い画面で起動するはずです。そして、場所が取得され、天気データが取得されると、次の画面である場所画面にプッシュする必要があります。

少し時間がかかりますよね？

そして、ユーザーがしばらくの間空白の画面を見つめているようなものです。

そのため、アプリがクラッシュしたか機能していないと思わせるかもしれません。

読み込みインジケーターを読み込み画面に追加して、バックグラウンドで何かが発生していることを実際に認識し、データの取得または場所の取得を待機します。使用するパッケージはFlutter SpinKitと呼ばれます。

また、アプリで使用できるさまざまなアニメーションローディングインジケータの束にアクセスできます。

そして私が特に気に入っているのは、ここでこの二重の境界線です。これは、何かが起こっていることを示し、それを待って、画面を遷移させます。

それでは、アプリの依存関係としてこれを追加しましょう。

pubspec.yamlに再度アクセスして、httpの下に追加します。そして、書式設定の一貫性を保つために通常どおり引用符を削除し、packages.getを実行してそのパッケージを取得します。

これが完了したら、ロード画面にインポートできるので、ここの一番上にあります。そして、ビルドメソッド内で使用します。

したがって、使用する方法は、必要なものが何であれ、SpinKitの回転円、SpinKitの二重境界、またはSpinKitのウェーブなど、特定のウィジェットを利用することです。

そして、色を設定し、サイズを設定できます。

それが私が足場の中でやろうとしていることです。それで、私のscaffoldは単に中心ウィジェットとなるボディを持つことになります。

したがって、SpinKitアニメーターを中央に配置でき、中央のウィジェットにはSpinKitDoubleBounceに設定する子プロパティがあります。 このウィジェットは、他のFlutterウィジェットと同じように使用できます。

いくつかのプロパティを追加するだけなので、colorプロパティを指定し、それを多分白に設定します。 また、比較的大きくするために100ピクセルに設定するサイズプロパティも指定します。

それでは、アプリを最初から再実行してみましょう。ホットリスタートすると、空白の画面から起動するはずです。

位置と天気を取得して画面にプッシュする前に、ロードインジケータをしばらく使用します。

そのため、アプリはよりユーザーフレンドリーになりました。 しかし、ロード画面に天気データがあるため、まだ問題がありますが、実際に必要なのはロケーション画面です。

それでは、どうすればそれを解析できますか？ ロード画面からロケーション画面へのデータの解析とそのデータの表示については、次のレッスンで取り上げます。

そこで会いましょう。